# CHAPTER 3

# Current Objective: Better Integration

There are theoretical and practical problems integrating models with simulations and with other models. The problems can appear to be simply software issues, but deeper theoretical issues often go hand-in-hand with these problems. We thus note a few of these problems in getting models to interact with simulations as well the basic problem of aggregating models.

## 3.1 Perception

At least since de Groot's early work (1946), perception has been deemed to play an essential role in cognition. Neisser (1976, p. 9) aptly summarizes it as "perception is where cognition and reality meet." This point of view has been buttressed in recent years with the emphasis given by Nouvelle AI (e.g., Brooks, 1992), which is based on reactive architectures, perceptual mechanisms, and on their coupling with motor behavior. Neuroscience (e.g., Kosslyn & Koenig, 1992) teaches that, due to evolutionary pressure, a large part of the brain deals with perception (mainly vision); hence, an understanding of perception is essential for understanding the behavior of combatants.

Perception-based behavior offers a series of advantages: it is fast, attuned to the environment, and optimized with respect to its coupling with motor behavior. However, its disadvantages include its tendency to be stereotyped and to lack generalization. In addition, from the point of view of the modeler, it is a difficult behavior to simulate well. This is in part due to the fact that low-level perception is still poorly understood (Kosslyn & Koenig, 1992), although recent progress in robotics and agent behavior give examples of successful implementation of basic perceptual mechanisms for use by cognition (e.g., Brooks, 1992; Zettlemoyer & St. Amant, 1999; and St. Amant & Riedl, 2001).

Perception may be seen as the common ground where various aspects of cognition meet, including motor behavior, concept formation and categorization, problem solving, memory, and emotions. In several of these domains, computer simulations illustrating the role of perception have been developed.

Brooks (1992) and others have investigated the role of perception in motor behavior with simple insect-like robots. The link between concept formation and (high-level) perception has been studied using the EPAM architecture (Gobet, Richman, Staszewski, & Simon, 1997). The role of perception in problem solving has been studied using Chunk Hierarchy and REtrieval Structures (CHREST), a variation of EPAM (Gobet, 1997; Gobet & Jansen, 1994) that also accounts for multiple memory regularities. Eye movements are simulated in detail in CHREST but not the low-level aspect of perception. (We will deal with the relation between problem solving and perception in Sec. 3.2.)

A more detailed simulation of low-level aspects of perception, such as feature extraction, is an important goal for the future of research on the relation of perception to other aspects of cognition. In addition, little work has been done on modeling perception in dynamically changing environments and on the effects of stress, emotion, motivation, and group factors on perception.

It is useful to separate *perception* from *cognition* in modeling human performance. The border between the model of the person and their environment can (arguably) be drawn at the boundary between cognition and perception, with perception belonging to a large extent in the environment model. This may be true for psychological reasons (Pylyshyn, 1999). It is also true to support tying models to simulations and for use of the resulting knowledge by cognition in problem solving (Ritter, Baxter, Jones, & Young, 2000). The typical acts performed by perception and motor action, such as determining the objects in view, their shapes and sizes, and then manipulating them, are most easily performed where the objects reside. This forces the implementation of theories of interaction into the simulation language instead of the modeling language.

It would be useful to have realistic stochastic distributions of differences in perception among individual agents, and also the ability to augment perception with instruments from field glasses to night sights. These devices could be modeled as *plug-ins* to the perception model. Models of perception in synthetic environments are typically simple, being a function of distance from observer to object (e.g., if there is a clear line of sight and the absence of cover and smoke). On the other hand, human vision changes in important ways with the ambient level of light and with the part of the retina on which an image falls. The edges of the retina are particularly sensitive to the detection of a moving object, while the fovea has the best resolution for identifying distant objects and is most sensitive to color. The distance at which an object can be seen depends on its brightness, its size, and its contrast to the background as well as the permeability of the air to light. Thus, a detonation will be visible from a much greater range than a moving tank, which in turn will be much easier to spot than a motionless, camouflaged soldier.

*Situation awareness* is a term that is still the subject of much debate in the human factors and ergonomics communities (e.g., see the Special Issue of *Human Factors,* Volume 37, Issue 1). Pew and Mavor (1998) consider situation awareness to be a key concept in the understanding of military behavior. We agree, but also believe that situation awareness should be modeled at a finer level of detail than is currently often done (see Pew & Mavor, 1998, chap. 7, for a current review).

## 3.2 Combining Perception and Problem Solving

Pew and Mavor (1998) note that an important constraint on problem solving is perception, but do not explore this in detail. As mentioned in our discussion on expertise, perception plays an important role in skilled behavior—experts sometimes literally *see* the solution to a problem (de Groot, 1946/1978).

We may use Kosslyn and Koenig's (1992) definition: higher-level visual processing involves using previously stored information; lower-level visual processing does not involve such stored information and is driven only by the information impinging on the retina. We focus here on higher-level perception and, thus, we will not consider mechanisms used for finding edges, computing depth, and so on.

Neisser's *Cognition and Reality* (1976) describes what is often referred to as the perceptual cycle. This approach underpins a vast amount of the cognitive engineering literature and research. At its simplest, the perceptual cycle is a cycle between the exploration of reality and representing this reality as schemas (in the general sense). Schemas direct exploration (perceptual, haptic, etc.) that involves sampling the object (looking at the real world), which may alter the object, which means that the schemas have to be modified. (See Neisser, 1976, p. 21, or p. 112 for a more complete description.) This work suggests that an important aspect of behavior has been missing from many theories and models of problem solving that have not included perception.

It is natural that researchers have attempted in recent years to combine perception and problem solving in artificial systems. One can single out three main approaches: robotics, problem-solving architectures incorporating perception, and perceptual architectures being extended to problem solving.

In robotics, Nouvelle AI has attempted to build robots able to carry simple problem-solving behavior without explicit planning by linking sensor and motor abilities tightly (e.g., the behavior-based architecture of Brooks, 1992). Robots based on this approach are excellent at obstacle-avoiding behavior. It is, however, unclear how far this approach can be extended to more complex problem solving without incorporating some sort of planning.

Including perception in behavioral models is a useful way to add natural competencies and limitations to behavior. Pew and Mavor note that there are few models of how perception influences problem solving. Their summary can be extended and revised in this area, however. We have seen in existing cognitive models (Byrne, 2001; Chong, 2001; de Groot & Gobet, 1996; Gobet, 1997; Jones, Ritter, & Wood, 2000; Ritter & Bibby, 2001; Salvucci, 2001) and in AI models (Elliman, 1989; Grimes, Picton, & Elliman, 1996; St. Amant & Riedl, 2001) that perception is linked to and can provide behavioral competencies and restrictions on problem solving. While Pew and Mavor note that they are unaware of any attempt in Soar to model the detailed visual perceptual processes in instrument scanning (Pew & Mavor, 1998, p. 181), such models exist (Aasman, 1995; Aasman & Michon, 1992; Bass et al., 1995), and some are even cited by Pew and Mavor (1998, p. 95) for other reasons.

The Soar model reported by Bass et al. (1995) scans a simple air-traffic control display to find wind velocity. The model learns (chunks) this information and uses it and the display to track and land a plane through airport air traffic control. The model then reflects on what it did to consider a better course of action. This model shows tentative steps towards using Soar's learning mechanism for situation learning and assessment based on information acquired through active perception (see Pew & Mavor, 1998, p. 197). Modeling visual cognition within Soar is ongoing at the University of Southern California's Information Sciences Institute (USC/ISI; Hill, 1999) and at the Pennsylvania State University.

The EPAM architecture (Feigenbaum & Simon, 1984), the initial goal of which was to model memory and perception, has recently been extended into a running production system (Gobet & Jansen, 1994; Lane, Cheng, & Gobet, 1999). The chunks learned while interacting with the task environment can later be used as conditions of productions. The same chunks are also used for the creation of schemas and for directing eye movements.

Recently, there have been several attempts to move the perception component from models into the architectures, regularizing and generalizing the results in the process. Prominent cognitive architectures Soar and ACT-R have been extended to incorporate perceptual modules, and PSI also has a perceptual module. With Soar, a perceptual module is available based on EPIC (Chong & Laird, 1997) and another based loosely on a spotlight theory of attention (Ritter et al., 2000). With ACT-R, two perceptual modules have been developed independently: the Nottingham architecture (Ritter et al., 2000) and ACT-R/PM (based on but also extending EPIC; Byrne, 1997, 2001). This approach creates situated models of cognition, that is, models that interact with (simulations of) the real world.

None of these approaches has been tested with complex, natural, and dynamically changing environments. The robotics approach is the only one currently demonstrated to cope with natural, albeit rather simple, environments. The two other approaches can interact with computer interfaces that are complex and dynamic (e.g., Salvucci, 2001).

## 3.3 Integration of Psychology Theories

A glance at almost any psychology textbook reveals that the study of human cognition is conventionally divided into topics that are presented as if they have little to do with each other. There will be separate chapters on attention, memory, problem solving, and so on. However, the range and variety of tasks undertaken by people at work, and also those tackled by synthetic agents, typically require the application and interplay of many different aspects of cognition simultaneously or in close succession. Interacting with a piece of electronic equipment, for example, can draw upon an agent's capacity for perception, memory, learning, problem solving, motor control, decision making, and many more capabilities. The question of how to integrate these different facets of cognition is therefore an important one for the simulation of human behavior.

Integrating theories across different topics of cognition is an issue that has rarely been addressed directly and provides an important focus for future work. Agents in synthetic environments (e.g., R. Jones, Laird, Nielsen, Coulter, Kenny, & Koss, 1999) implicitly integrate multiple aspects of behavior. What research exists has been carried out, appropriately enough, under the heading of unified theories of cognition using architectures such as Soar and ACT-R. Soar offers a promising basis for such integration. Its impasse-driven organization enables it to access different areas of cognitive skill as the need arises, and its learning mechanism (which depends on cognitive processing in those impasses) enables relevant information from the different areas to be integrated into directly applicable knowledge for future use. ACT-R also integrates multiple components.

## 3.4 Integration and Reusability of Models

Integration of theories can be also viewed as integration of models as software, sometimes called *reuse*. It has been true for years that reuse is important; this is true for two fundamental reasons. First, reuse saves effort. In the field of object-oriented software development, figures are often quoted for the costs associated with development with reuse in mind. The extra time spent in initial development is something like 20%. When the code is reused, an application can be created in 40% of the development time for new code. Second, and perhaps more importantly in these domains, reuse ensures consistency across simulations and time, particularly important when creating unified theories of cognition.

There are also serious problems restricting the reuse of cognitive models. Cognitive models are not generally reused, even when they have been created in a cognitive architecture designed to facilitate their reuse. There are exceptions. Pearson's Version 2 of his Symbolic Concept Acquisition model and its explanatory displays is an exception (available at ai.eecs.umich.edu/soar/soar-group.html). Other exceptions include PDP toolkits such as O'Reilly's PDP++ (www.cnbc.cmu.edu/PDP++/PDP++.html). But, overall, cognitive modeling does not have the level of system reuse and visual displays that the AI and expert systems communities now take for granted. This problem is being noticed by others as well (Wray, 2001).

There are some examples of reuse that should be emulated and expanded. ACT-R now maintains a library of existing models (act.psy.cmu.edu). We have found that the mere existence of a library of student models (www.nottingham.ac.uk/pub/soar/nottingham/) has led to increasingly better student projects. Work by Young (1999) on building a zoo of runnable cognitive models is another example of such use done broadly. There is little reason to believe that these results would not scale up. These improvements to the modeling environment have helped move learning Soar (Ritter & Young, 1999) and ACT-R (Anderson & Lebiere, 1998) from being a lengthy apprenticeship to being something that can be taught in undergraduate courses.

Such integration is illustrated most clearly in a model of natural language sentence processing (Lewis, 1993), in which lexical, syntactic, semantic, pragmatic, and domain-specific knowledge are brought together in learned rules (Soar chunks) to guide language comprehension. Probably the model that has gone furthest in demonstrating this kind of integration is the cognitive model of the NASA Test Director, the person responsible for coordinating the preparation and launch of the space shuttle. Nelson, Lehman, and John (1994) describe a Soar model of a fragment of the Test Director's performance, which incorporates problem solving, listening to audio communications, understanding language, speaking, visual scanning (through a procedure manual), page turning, and more. Such integrated models are also starting to be created in ACT-R (Anderson & Lebiere, 1998).

Integration of a slightly different flavor—across capabilities rather than across textbook-like topics of cognition—is illustrated in another Soar model, this one being of exploratory learning of an interactive device (Rieman et al., 1996). At first glance, it might seem that exploratory learning is not especially relevant to the human behavior that is, apart from questions of training, the main focus of this report. Fighter pilots and tank commanders are highly trained and expert individuals, and presumably do not learn significantly from further experiences. However, component skills such as comprehending a novel situation, looking around to discover relevant options, and assessing a course of action—which are fundamental components of expert skill—are also precisely what are required for exploratory learning and reactive planning in uncertain environments.

Rieman et al. (1996) describe the IDXL model, which models an experienced computer-user employing exploratory learning to discover how to perform specified tasks with an unfamiliar software application. IDXL searches both the external space provided by the software and the internal space of potentially relevant knowledge. It seeks to comprehend what it finds and approximates the *rationally optimal* strategy (Anderson, 1990) for exploratory search. A typical sequence of interrelated capabilities would be for the model first to learn how to start a spreadsheet program from external instruction; then to use that new knowledge as a basis for analogy to discover how to start a graph-drawing package; and then to build on its knowledge by learning through exploration how to draw a graph. The model works with a limited working memory, employs recognition-based problem solving (Howes, 1993), and acquires display-based skill (Payne, 1991) in an interactive, situated task.

These problems of reusability are even more acute when creating models for synthetic environments because of the size and type of models. This is true for several reasons: the knowledge is more extensive and exact than many laboratory domains previously studied. The models must interact with complex, interactive simulations. The work may be classified, which will add an additional constraint in hiring someone with multiple skills. Scenarios may simulate hours of behavior rather than the minutes of typically modeled laboratory tasks. This represents a lot of knowledge, and the timeframe can make troubleshooting more difficult. Finally, there are many cases where an explanation facility is required to explain the model's behavior for other observers.

## 3.5 Summary

A framework to assist with integration and reuse will have to be developed. It should be common in the sense that the appropriate simulation entities and analysis tools would be available, and for a given application or analysis, developers would plug them together. The DIS protocol and ModSAF are being used in this way to some extent, but they are hard to use and do not support the desired level of ease of use nor the level of cognitive realism.